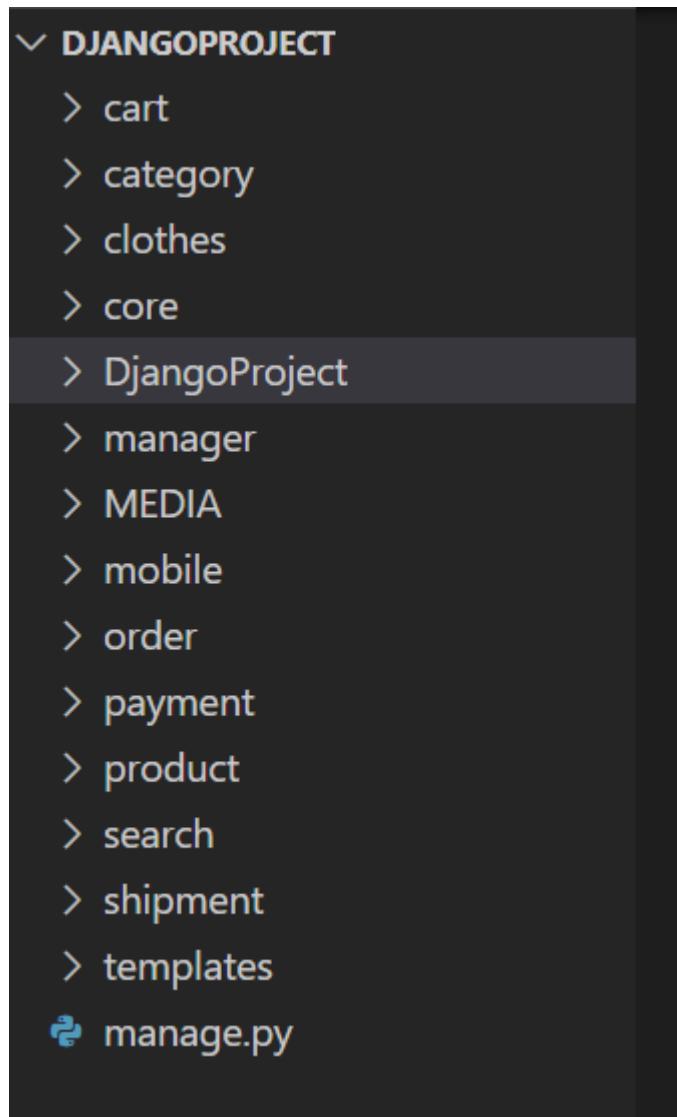


Phần 1 Tạo các app, thiết kế models và kết nối với database:

Các app đã được tạo



Thêm các app vào INSTALLED_APPS trong file setting.py

```
INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'core',
    'category',
    'product',
    'cart',
    'search',
    'clothes',
    'mobile',
    'order',
    'payment',
    'shipment',
    'manager',]
```

Thiết lập DATABASES trong file setting.py

```
DATA BASES = {  
    "default": {  
        "ENGINE": "django.db.backends.mysql",  
        "NAME": 'store',  
        "USER": "root",  
        "PASSWORD": "12345",  
        "HOST": "127.0.0.1",  
        "PORT": "3306",  
    },  
    'mongodb': {  
        "ENGINE": "djongo",  
        'name': 'store',  
        'port': 27017  
    }  
}
```

app Cart

```

from django.db import models
from django.contrib.auth.models import User

from product.models import Product

class Cart(models.Model):
    date_added = models.DateTimeField(auto_now_add=True)
    user = models.ForeignKey(User, on_delete=models.CASCADE)

    class Meta:
        app_label = 'cart'
        db_table = 'cart'
        ordering = ['-date_added']
        verbose_name_plural = 'Carts'

    def __str__(self):
        return str(self.user.username)

class CartProduct(models.Model):
    cart = models.ForeignKey(Cart, related_name='cartproducts', on_delete=models.CASCADE)
    product = models.ForeignKey(Product, on_delete=models.CASCADE)
    quantity = models.IntegerField(default=1)

    class Meta:
        app_label = 'cart'
        db_table = 'cart_product'
        verbose_name_plural = 'CartProducts'

```

app clothes

```

from django.db import models

from category.models import Category

class Clothes(models.Model):
    name = models.CharField(max_length=255, unique=True)
    slug = models.SlugField(max_length=255, unique=True,
                           help_text='Unique value for product page URL, created from name.')
    price = models.IntegerField(default=0)
    designer = models.CharField(max_length=255)
    old_price = models.IntegerField(default=0)
    image = models.ImageField(upload_to='product_images', blank=True, null=True)
    is_active = models.BooleanField(default=True)
    is_bestseller = models.BooleanField(default=False)
    description = models.TextField(null=True)
    categories = models.ManyToManyField(Category, related_name='clothes')

    class Meta:
        app_label = 'clothes'
        db_table = 'clothes'
        verbose_name_plural = 'Clothes'

    def __str__(self):
        return self.name

```

app category

```
from django.db import models

class Category(models.Model):
    name = models.CharField(max_length=255)
    slug = models.SlugField(max_length=255, unique=True,
                           help_text='Unique value for product page URL, created from name.', null=True)
    description = models.TextField(null=True, blank=True)
    is_active = models.BooleanField(default=True)
    created_at = models.DateTimeField(auto_now_add=True)
    updated_at = models.DateTimeField(auto_now=True)

    class Meta:
        app_label = 'category'
        db_table = 'category'
        ordering = ['-created_at']
        verbose_name_plural = 'Categories'

    def __str__(self):
        return self.name
```

app mobile

```
from django.db import models

from category.models import Category

class Mobile(models.Model):
    name = models.CharField(max_length=255, unique=True)
    slug = models.SlugField(max_length=255, unique=True,
                           help_text='Unique value for product page URL, created from name.', null=True)
    price = models.IntegerField(default=0)
    designer = models.CharField(max_length=255)
    old_price = models.IntegerField(default=0)
    image = models.ImageField(upload_to='product_images', blank=True, null=True)
    is_active = models.BooleanField(default=True)
    is_bestseller = models.BooleanField(default=False)
    description = models.TextField(null=True)
    categories = models.ManyToManyField(Category, related_name='mobiles')

    class Meta:
        app_label = 'mobile'
        db_table = 'mobile'
        verbose_name_plural = 'Mobiles'

    def __str__(self):
        return self.name
```

app product

```

from django.db import models

from category.models import Category

class Product(models.Model):
    name = models.CharField(max_length=255, unique=True)
    slug = models.SlugField(max_length=255, unique=True,
                           help_text='Unique value for product page URL, created from name.')
    author = models.CharField(max_length=255, null=True)
    publisher = models.CharField(max_length=255, null=True)
    price = models.IntegerField(default=0)
    old_price = models.IntegerField(default=0)
    image = models.ImageField(upload_to='product_images', blank=True, null=True)
    is_active = models.BooleanField(default=True)
    is_bestseller = models.BooleanField(default=False)
    description = models.TextField(null=True)
    created_at = models.DateTimeField(auto_now_add=True)
    updated_at = models.DateTimeField(auto_now=True)
    categories = models.ManyToManyField(Category, related_name='products')

    class Meta:
        app_label = 'product'
        db_table = 'product'
        ordering = ['-created_at']
        verbose_name_plural = 'Products'

    def __str__(self):
        return self.name

```

Phần 2. Tạo Rest API để kết nối với các dịch vụ

hiển thị tất cả các product, clothes, mobile
các file urls trong từng app tương ứng

```

from django.contrib import admin
from django.urls import path

from . import views

urlpatterns = [
    path('', views.index, name='index'),
]

```

```
from django.shortcuts import render
from category.models import Category

from clothes.models import Clothes
from mobile.models import Mobile
from product.models import Product

def index(request):
    products = Product.objects.all()
    clothes = Clothes.objects.all()
    mobiles = Mobile.objects.all()
    categories = Category.objects.all()
    return render(request, 'index.html', {
        'products': products,
        'clothes': clothes,
        'mobiles': mobiles,
        'categories': categories,
    })
```

giao diện tương ứng

List Book



Nghiên cứu tâm lý trẻ em

Price: 1399

Nghiên cứu bệnh tâm thần

Price: 12

Đầu tư chứng khoán hiệu quả

Price: 13

List Clothes



Áo sơ mi trắng

Price: 9999

List Mobiles



Iphone 15

Price: 999

file html tương ứng:

```

{% extends 'base.html' %}

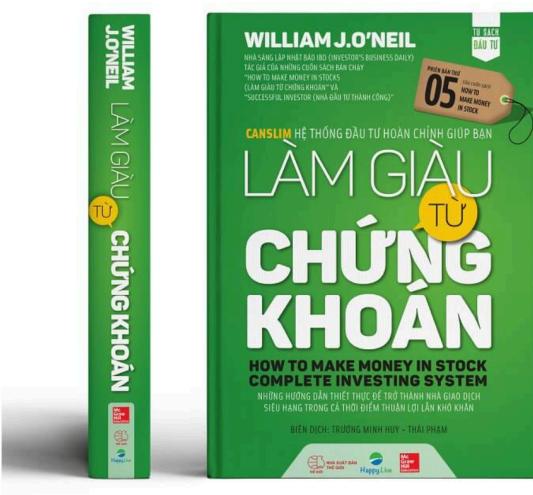
{% block title %}Welcome{% endblock %}

{% block content %}


## List Book</h2> <div class="grid grid-cols-3 gap-3"> {% for product in products %} <div> <a href="{% url 'product:detail' product.id %}"> <div>  </div> <div class="p-6 bg-white rounded-b-xl"> <h2 class="text-2xl">{{ product.name }}</h2> <p class="text-gray-500">Price: {{ product.price }}</p> </div> </a> </div> {% endfor %} </div> <h2 class="mb-12 text-2xl text-center">List Clothes</h2> <div class="grid grid-cols-3 gap-3"> {% for cloth in clothes %} <div> <a href="{% url 'clothes:detail' cloth.id %}"> <div>  </div> </div> {% endfor %} </div>


```

Hiển thị chi tiết product, clothes, mobile



Nghiên cứu tâm lý trẻ em

Price: 1399
 Author: Bùi Xuân Huấn
 Publisher: NXB Việt Nam
 Category: Tâm lý học
 Description: dsadsadsa

[Contact seller](#) [Add to cart](#)

```

from django.contrib import admin
from django.urls import path

from . import views

app_name = 'product'

urlpatterns = [
    path('<int:pk>', views.detail, name='detail'),
]

```

Phần 3 Tìm kiếm sản phẩm
click browse ở giao diện

Bookstore

New item Browse [Sign up](#) [Log in](#)

List Book



Nghiên cứu tâm lý trẻ em

Price: 1399



Nghiên cứu bệnh tâm thần

Price: 12



Đầu tư chứng khoán hiệu quả

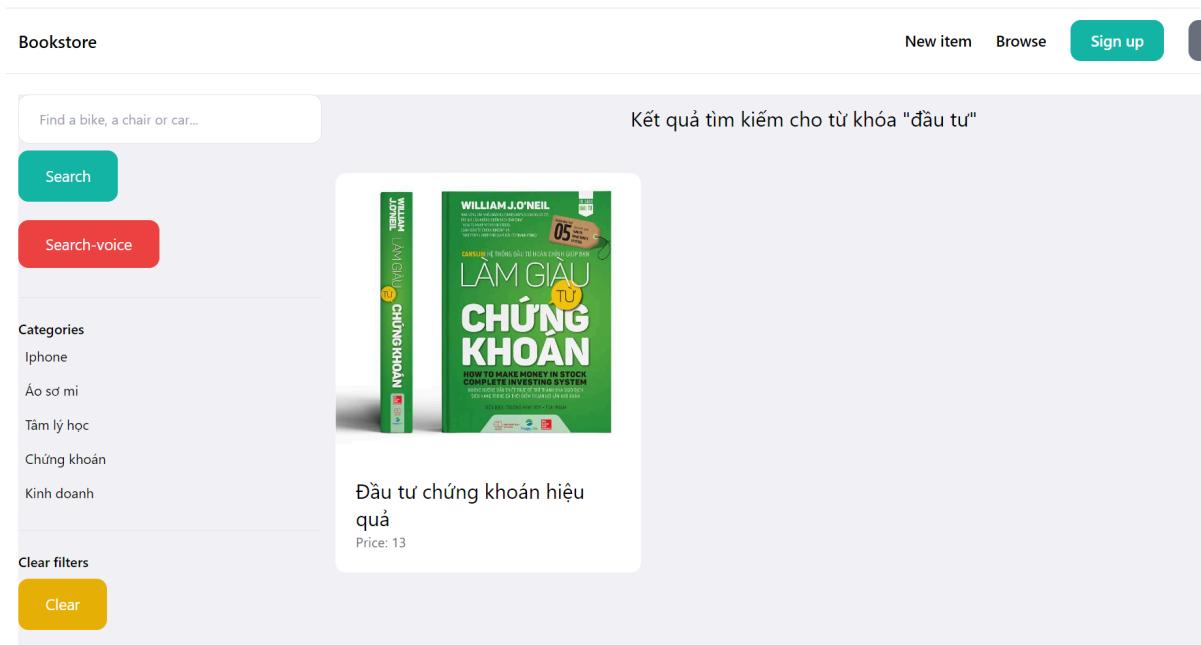
Price: 13

List Clothes

nhập từ khóa và click search



khi click tìm kiếm bằng voice, nhấp vào nút đỏ và nói từ khóa, hệ thống hiển thị danh sách thỏa mãn



quy trình
tại file view.py trong app search
đối với từ khóa nhập vào

```
def products(request):
    query = request.GET.get('query', '')
    category_id = request.GET.get('category', 0)
    categories = Category.objects.all()
    products = Product.objects.all()
    clothes = Clothes.objects.all()
    mobiles = Mobile.objects.all()

    if query:
        products = products.filter(Q(name__icontains=query))
        clothes = clothes.filter(Q(name__icontains=query))
        mobiles = mobiles.filter(Q(name__icontains=query))

    if category_id:
        category = Category.objects.get(pk=category_id)
        products = Product.objects.filter(categories=category)

    return render(request, 'search/products.html', {
        'products': products,
        'query': query,
        'clothes': clothes,
        'mobiles': mobiles,
        'categories': categories,
        'category_id': int(category_id),
    })
```

đối với xử lí tìm kiếm giọng nói

```
def productsByVoice(request):
    recognizer = sr.Recognizer()

    text = ''

    with sr.Microphone() as source:
        print("Đang nghe... (nói 'kết thúc' để dừng lại)")

        try:
            # Tự động điều chỉnh nền để loại bỏ tiếng ồn
            recognizer.adjust_for_ambient_noise(source)

            audio_data = recognizer.listen(source, timeout=2)
            print("Đã nghe xong. Đang nhận dạng...")

            text = recognizer.recognize_google(audio_data, language="vi-VN")
            print("Văn bản nhận dạng từ giọng nói: {}".format(text))

        except sr.UnknownValueError:
            print("Không thể nhận dạng giọng nói")
        except sr.RequestError as e:
            print("Lỗi khi gửi yêu cầu đến API Google: {}".format(e))
        except sr.WaitTimeoutError:
            print("Hết thời gian chờ. Không có âm thanh được nghe.")

    # Kiểm tra nếu text là None hoặc trống thì gán giá trị khoảng trắng
    text = text or ''

    products = Product.objects.all()
    clothes = Clothes.objects.all()
```

```
61
62     # Kiểm tra nếu text là None hoặc trống thì gán giá trị khoảng trắng
63     text = text or ''
64
65     products = Product.objects.all()
66     clothes = Clothes.objects.all()
67     mobiles = Mobile.objects.all()
68     products = products.filter(Q(name__icontains=text))
69     clothes = clothes.filter(Q(name__icontains=text))
70     mobiles = mobiles.filter(Q(name__icontains=text))
71     categories = Category.objects.all()
72     category_id = request.GET.get('category', 0)
73
74     return render(request, 'search/products.html', {
75         'products': products,
76         'text': text,
77         'clothes': clothes,
78         'mobiles': mobiles,
79         'categories': categories,
80         'category_id': int(category_id),
81     })
82
```

cấu hình urls tương ứng

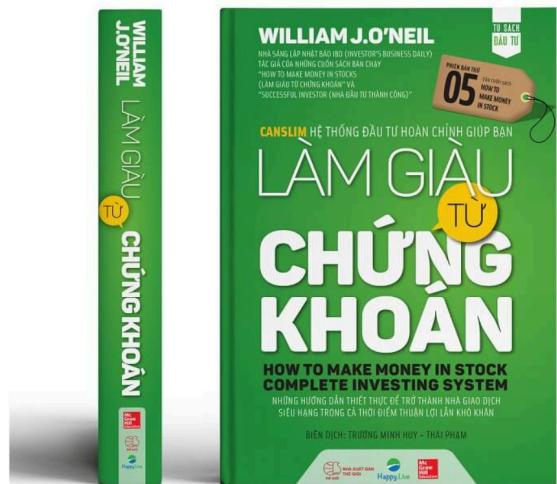
```
from django.contrib import admin
from django.urls import path

from . import views

app_name = 'search'

urlpatterns = [
    path('', views.products, name='products'),
    path('voice//', views.productsByVoice, name='products-voice'),
]
```

ĐÓi với add to cart
click add to cart



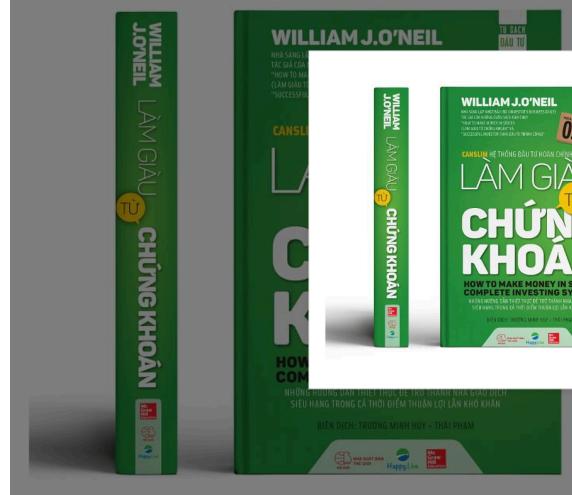
Nghiên cứu tâm lý trẻ em

Price: 1399
Author: Bùi Xuân Huân
Publisher: NXB Việt Nam
Category: Tâm lý học
Description: dsadsadsa

Contact seller

Add to cart

nhập số lượng và submit



Nghiên cứu tâm lý trẻ em

Số lượng: 4

Add to cart

hệ thống lưu vào giỏ hàng

Carts



Nghiên cứu tâm lý trẻ em

Đơn giá 1399 USD

Số lượng: 17

Thành tiền: 23783 USD



Đầu tư chứng khoán hiệu quả

Đơn giá 13 USD

Số lượng: 14

Thành tiền: 182 USD



Nghiên cứu bệnh tâm thần

Đơn giá 12 USD

Số lượng: 9

Thành tiền: 108 USD

Tổng tiền: 24073 USD

quy trình

tạo view tương ứng với add to cart và xem cart

```
from cart.models import Cart, CartProduct
from product.models import Product

@login_required
def add_to_cart(request, pk):
    product = get_object_or_404(Product, pk=pk)
    quantity = int(request.POST.get('quantity', 1))
    cart, created = Cart.objects.get_or_create(user=request.user)
    cart_product, product_created = CartProduct.objects.get_or_create(cart=cart, product=product)
    cart_product.quantity += quantity
    cart_product.save()
    return redirect('product:detail', pk=pk)

@login_required
def carts(request):
    cart = Cart.objects.filter(user=request.user).first()
    tongtien = 0
    for cproduct in cart.cartproducts.all():
        tongtien += cproduct.product.price * cproduct.quantity
    return render(request, 'cart/carts.html', {
        'cart': cart,
        'tongtien': tongtien,
    })
```

cấu hình urls

```
from django.urls import path

from . import views

app_name = 'cart'

urlpatterns = [
    path('<int:pk>/add-to-cart/', views.add_to_cart, name='add_to_cart'),
    path('carts/', views.carts, name='carts'),
]
```